Pertemuan 2

Struktur Kontrol Percabangan

Objektif :

1. Mahasiswa dapat memahami konsep struktur kontrol percabangan dalam pemrograman.
2. Mahasiswa dapat menggunakan struktur kontrol pemilihan (if, else, switch) yang digunakan.
3. Menggunakan pernyataan-pernyataan percabangan (break, continue, return) yang digunakan untuk mengatur arah dari aliran program.
4. Mahasiswa dapat membuat program yang berisi alur program secara bercabang melalui contoh kasus.

P2.1 Teori

1. Struktur Kontrol Percabangan

Struktur kontrol percabangan adalah pernyataan dari Java yang mengijinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain. Jenis Percabangan pada Java terdiri dari :

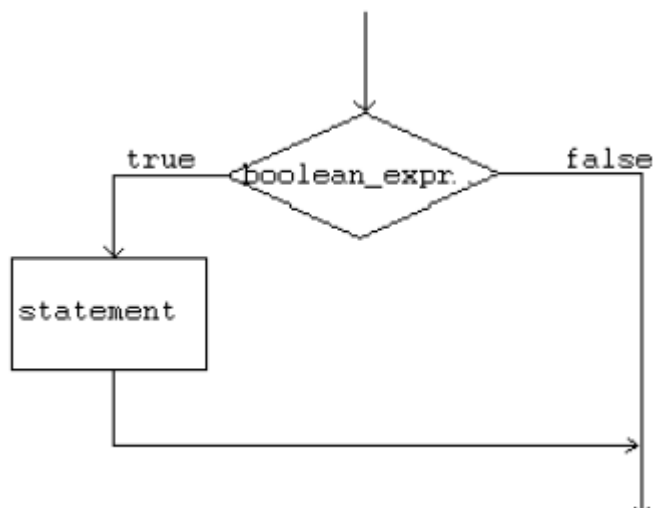
1. Statement If
2. Statement If-else
3. Statement If-else-if
4. Statement Switch case

1.1 Statement If

Pernyataan *if* akan menentukan sebuah pernyataan (atau blok kode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar*(true).*

Bentuk dari pernyataan if :

```
if( boolean_expression )
statement;
atau
if( boolean_expression ){
statement1;
statement2;
. . .
}
```

Flowchart Statement If yaitu :

Berikut ini adalah potongan kode dari pernyataan if :

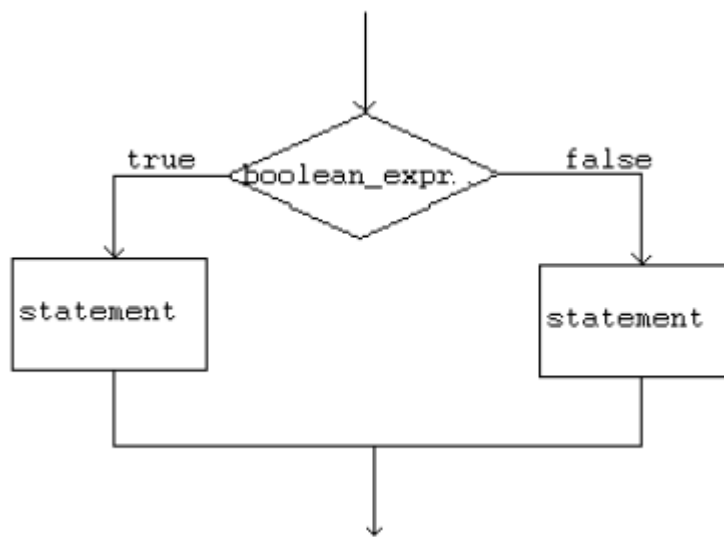
```
int grade = 68;
if( grade > 60 ) System.out.println("Congratulations!");
atau
int grade = 68;
if( grade > 60 ){
System.out.println("Congratulations!");
System.out.println("You passed!");
}
```

1.2 Statement If else

Pernyataan *if-else* digunakan apabila kita ingin mengeksekusi beberapa pernyataan dengan kondisi *true* dan pernyataan yang lain dengan kondisi *false*.

Bentuk statement if-else :

```
if( boolean_expression )
statement;
else
statement;
dapat juga ditulis seperti,
if( boolean_expression ){
statement1;
statement2;
. . .
}
else{
}
statement1;
statement2;
. . .
```

Flowchart Statement if-else yaitu :

Berikut ini adalah potongan kode dari pernyataan if-else :

```
int grade = 68;
if( grade > 60 ) System.out.println("SELAMAT!");
else System.out.println("MAAF SALAH!");
atau
int grade = 68;
if( grade > 60 ){
System.out.println("SELAMAT!");
System.out.println("ANDA LOLOS!");
}
else{
}
System.out.println("MAAF SALAH!");
```

1.3 Statement if-else-if

Pernyataan pada bagian kondisi *else* dari blok *if-else* dapat menjadi struktur *if-else* yang lain. Kondisi struktur seperti ini mengijinkan kita untuk membuat seleksi persyaratan yang lebih kompleks.

Bentuk statement if-else-if :

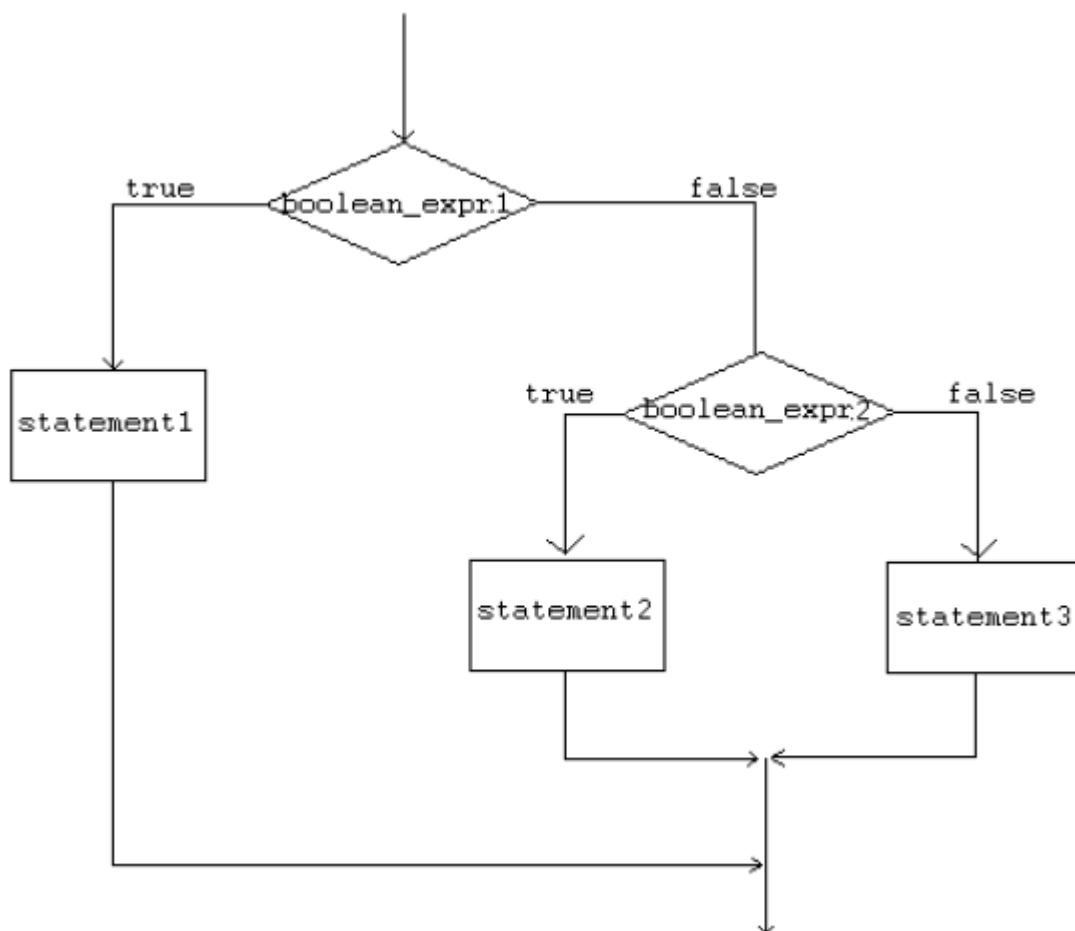
```
if( boolean_expression1 )
statement1;
else if( boolean_expression2 )
```

statement2;

else

statement3;

Sebagai catatan : Anda dapat memiliki banyak blok else-if sesudah pernyataan *if*. Blok *else* bersifat opsional dan dapat dihilangkan. Pada contoh yang ditampilkan di atas, jika *boolean_expression1* bernilai *true*, maka program akan mengeksekusi *statement1* dan melewati pernyataan yang lain. Jika *boolean_expression2* bernilai *true*, maka program akan mengeksekusi *statement2* dan melewati *statement3*.

Flowchart Statement If else if yaitu :

Berikut ini adalah potongan kode dari pernyataan if else if :

```
int grade = 68;
if( grade > 90 ){
System.out.println("Sangat Baik!");
}
```

```
else if( grade > 60 ){
System.out.println("Sangat Baik!");
}
else{
}
System.out.println("Maaf Gagal!");
```

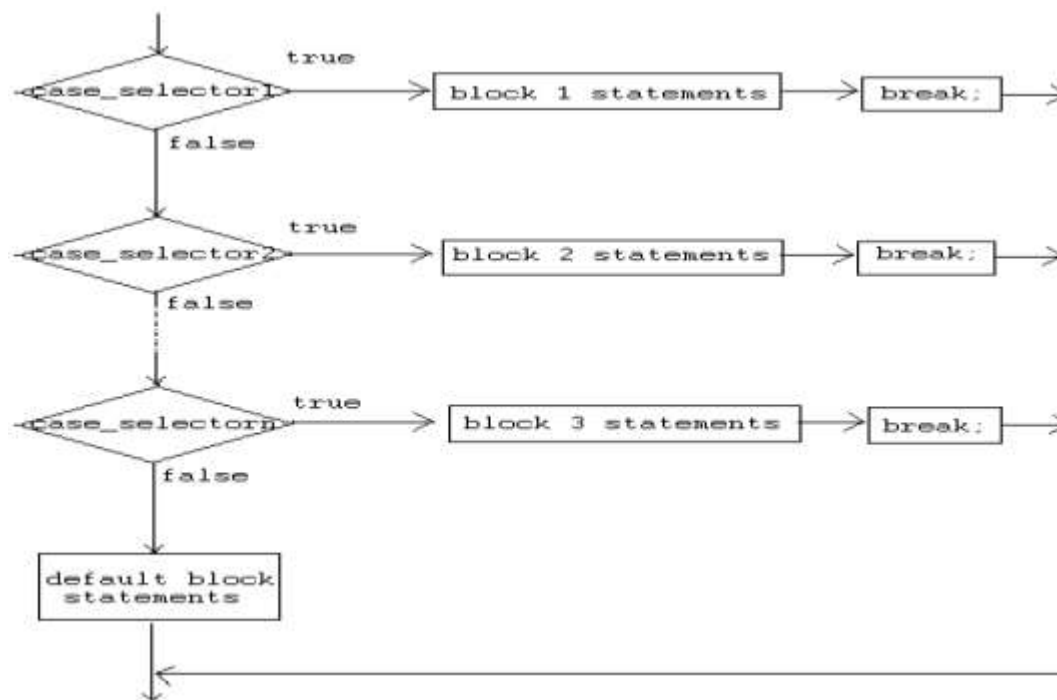
1.4 Statement Switch Case

Penyataan percabangan kedua yang dimiliki Java adalah **switch**. Pernyataan **switch** lebih jarang digunakan, tetapi sering bermanfaat apabila kita ingin menuliskan percabangan multi arah. Pernyataan switch memiliki bentuk sebagai berikut :

```
switch (ekspresi) {
  case nilai1:
    perintah1
    break;
  case nilai2:
    perintah2
    break;
  case nilai3:
    perintah3
    break;
  default:
    perintah_lain
}
```

Di sini pernyataan switch akan mencari nilai ekspresi yang sesuai dengan nilai-nilai yang didaftarkan pada pernyataan **case**. Jika salah satu nilai ditemui, maka program akan melompat ke cabang **case** tersebut dan melakukan perintah yang terdapat di sana. Jika tidak ditemui, maka program akan melompat ke perintah yang terdapat pada pernyataan **default**.

Catatan ekspresi hanya bisa berbentuk nilai bilangan bulat (int, short, dan sejenisnya) atau karakter, sehingga kita tidak bisa menggunakan switch untuk mengevaluasi ekspresi yang berbentuk String.

Flowchart Statement Switch Case yaitu :

Berikut ini adalah potongan kode dari pernyataan Switch case :

```
public class Grade
{
public static void main( String[] args )
{
int grade = 92;
switch(grade){
case 100:
System.out.println( "Excellent!" );
break;
case 90:
case 80:
default: }
}
}
System.out.println("Good job!" );
break;
System.out.println("Study harder!" );
break;
System.out.println("Sorry, you failed.");
```

2. Struktur Kontrol Percabangan Tidak Lokal

Bahasa Java menyediakan beragam kendali percabangan tidak lokal, yaitu: *break*, *return*, dan *continue*.

2.1. Statement Break

Java tidak memiliki pernyataan **goto**. Penggunaan **goto** di bahasa pemrograman lain adalah cara untuk mencabang secara sembarang, yang membuat program sulit untuk dimengerti dan mengurangi optimasi *compiler* tertentu. Namun, ada beberapa keadaan dimana **goto** berguna dan bentuk yang sah untuk pengaturan program.

Pernyataan **break** pada Java dirancang untuk mengatasi semua kasus tersebut. Istilah **break** mengacu pada proses memecahkan blok program. Proses tersebut memerintahkan *runtime* untuk menjalankan program di belakang blok tertentu. Untuk dapat ditunjuk, suatu blok diberi nama, dan Java memiliki bentuk label untuk menyatakan nama suatu blok.

Contoh:

```
class Break
{
public static void main(String[] args)
{
boolean t =true;
a: {
b: {
c: {
System.out.println("Sebelum break");
if (t)
break b;
System.out.println("Ini tidak akan dieksekusi");
}
System.out.println("Ini tidak akan dieksekusi");
}
System.out.println("Ini adalah setelah b");
}
}
}
```

Hasil (*output*) dari contoh listing program diatas:

Sebelum break

Ini adalah setelah b

2.2. Statement Return

Java menggunakan bentuk sub-rutin yang disebut *method* untuk mengimplementasikan antarmuka procedural ke kelas objek. Setiap saat dalam *method* dapat digunakan pernyataan **return** yang menyebabkan eksekusi mencabang kembali ke pemanggil *method*.

Contoh:

```
class Return1
{
public static void main(String[] args)
{
boolean t = true;
System.out.println("Before the return");
if (t)
return;
System.out.println("This won't execute");
}
}
```

Hasil (*output*) dari contoh listing program diatas:

Before the return

2.3. Statement Continue

Seringkali kita ingin keluar lebih cepat dari perulangan. Kita mungkin juga ingin meneruskan perulangan, tetapi harus menghentikan sisa proses pada program untuk iterasi yang bersangkutan. Ini dilakukan dengan **goto** yang memintas program, tetapi masih di dalam perulangan. Pernyataan *continue* di Java melakukan persis seperti itu.

Berikut contoh program penggunaan *continue* yang menyebabkan 2 bilangan dicetak dalam setiap baris:

```
class Continue1
{

public static void main(String[] args)
{
for (int i = 0; i<10; i++) {
System.out.print(i + " "); if
(i% 2 == 0)
continue;
System.out.println("");
}
}
}
```

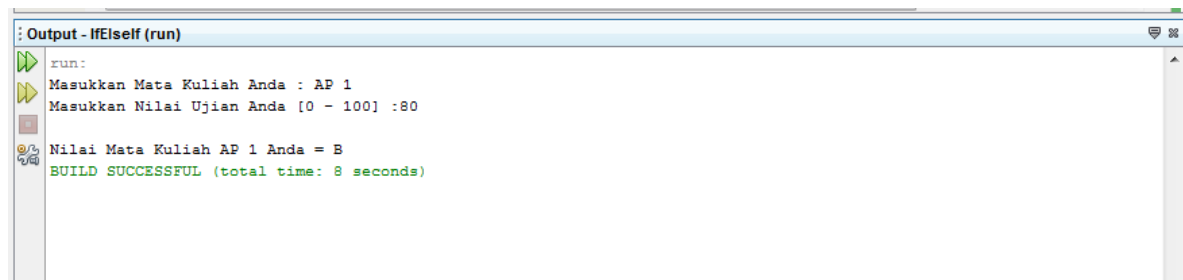
Hasil (*output*) dari contoh listing program di atas:

0 1

2 3

4 5

6 7

8 9

P2.2 Contoh Kasus

Buat program untuk menentukan nilai yang diperoleh mahasiswa. Penilaian dimasukkan dalam bentuk angka dan output nilai dalam bentuk huruf (A-E). Kondisi yang di gunakan:

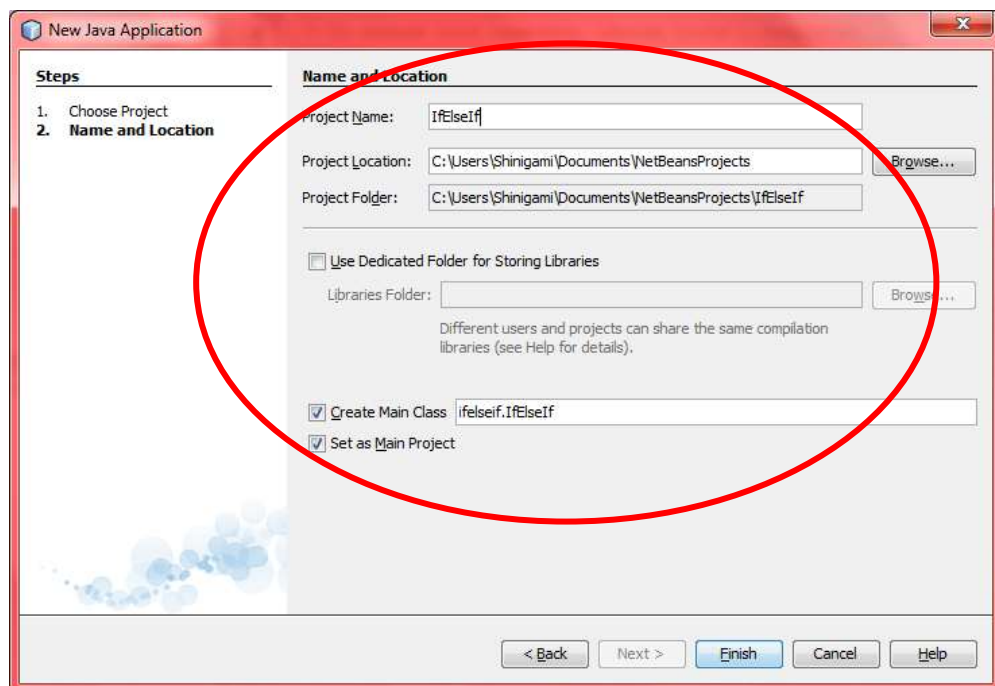
- Output nilai “A”, jika nilai antara 90 – 100
- Output nilai “B”, jika nilai 80 – 89
- Output nilai “C”, jika nilai 60 – 79
- Output nilai “D”, jika nilai 50 – 59
- Jika nilai kurang dari 50, maka menghasilkan output nilai “E”

Output yang diinginkan seperti gambar di bawah ini:

```
Output - IfElseIf (run)
run:
Masukkan Mata Kuliah Anda : AP 1
Masukkan Nilai Ujian Anda [0 - 100] :80

Nilai Mata Kuliah AP 1 Anda = B
BUILD SUCCESSFUL (total time: 8 seconds)
```

Langkah-langkah pengerjaan:

1. Jalankan Netbeans Anda.
2. Lalu buat file project baru dengan memilih menu File – New Project, atau dengan menggunakan hotkey Ctrl+Shift+N.
3. Pilih jenis project yang akan dibuat (Java – Java Application)
4. Tentukan nama project dan lokasi penyimpanan project. Contoh: nama project: IfElseIf, dan nama kelas: IfElseIf

5. Ketikkan kode program di bawah ini pada code editor

```
package ifelseif;

import java.io.*; //library untuk melakukan inputan
public class IfElseIf { /*kelas utama otomatis terbuat
                        saat membuat new file pada netbeans */

    public static void main(String[] args) throws Exception{ //main method
        // TODO code application logic here
        DataInputStream masuk = new DataInputStream(System.in); /*deklarasi variabel masuk
                                            sebagai inputan utama */

        String strnilai = null; // deklarasi variabel-variabel yang akan digunakan
        String matkul = null;

        try{
            System.out.print("Masukkan Mata Kuliah Anda : ");
            matkul = masuk.readLine(); // membaca dan menyimpan nilai inputan ke variabel
matkul
            System.out.print("Masukkan Nilai Ujian Anda [0 - 100] :");
            strnilai = masuk.readLine();
            System.out.println();
        }catch(IOException ioe){}

        int nilai; // deklarasi variabel tipe bilangan

        /* mengubah input dari variabel strnilai dengan tipe data String
        ke variabel nilai dengan tipe data integer */
        nilai = Integer.parseInt(strnilai);

        if(nilai >=90 && nilai <=100) /* jika nilai yang diinput >= 90 dan <=100
                            maka statement dibawahnya akan dieksekusi.
                            jika tidak, lanjut ke seleksi kondisi else if berikutnya  */
        System.out.println("Nilai Mata Kuliah "+matkul+" Anda = A");
        else if(nilai >=80 && nilai <=89)
        System.out.println("Nilai Mata Kuliah "+matkul+" Anda = B");
        else if(nilai>=60 && nilai <=79)
        System.out.println("Nilai Mata Kuliah "+matkul+" Anda = C");
        else if(nilai >= 50 && nilai <=59)
        System.out.println("Nilai Mata Kuliah "+matkul+" Anda = D");
        else
        System.out.println("Nilai Mata Kuliah "+matkul+" Anda = E");
        }
    }
```

6. Build project tersebut dengan memilih menu Run – Build Main Project, atau dengan menggunakan hotkey F11

7. Jika tidak ada kesalahan, jalankan project tersebut dengan memilih menu Run – Run Main Project, atau dengan menggunakan hotkey F6.

P2.3 Latihan

Buatlah program untuk menghasilkan output sebagai berikut:

```
Output - SwitchCoba (run)
run:
Kedai Kita-Kita
Makanan
         1. Bakso
         2. Mie Ayam
         3. Bihun
Minuman
         1. Es Campur
         2. Soda
         3. Air Putih

Masukkan Pilihan Makanan [1 - 3] :2
Masukkan Pilihan Minuman [1 - 3] :2

Makanan yang Anda pesan adalah Mie Ayam
Minuman yang Anda pesan adalah Soda
BUILD SUCCESSFUL (total time: 7 seconds)
```

Jawaban:

1. Jalankan Netbeans Anda
2. Lakukan langkah-langkah pengerjaan seperti contoh kasus sebelumnya.
3. Pada code editor Netbeans, ketikkan program berikut:

```
package switchcoba;

import java.io.*;
public class SwitchCoba {

    public static void main(String[] args) throws Exception{
        // deklarasi variabel masuk sebagai inputan utama
        BufferedReader masuk = new BufferedReader(new
InputStreamReader(System.in));

        String pilmakanan = null;
        String pilminuman = null;

        try{
            System.out.println("Kedai Kita-Kita");
            System.out.println("Makanan");
            System.out.println("\t 1. Bakso");
            System.out.println("\t 2. Mie Ayam");
            System.out.println("\t 3. Bihun");
            System.out.println("Minuman");
            System.out.println("\t 1. Es Campur");
            System.out.println("\t 2. Soda");
            System.out.println("\t 3. Air Putih");
            System.out.println();
            System.out.print("Masukkan Pilihan Makanan [1 - 3] :");
            pilmakanan = masuk.readLine();
            System.out.print("Masukkan Pilihan Minuman [1 - 3] :");
            pilminuman = masuk.readLine();
           }catch(IOException ioe){}
```

```
        int makanan;
        int minuman;

        makanan = Integer.parseInt(pilmakanan);
        minuman = Integer.parseInt(pilminuman);
        System.out.println();
        switch(makanan){
          case 1:
            System.out.println("Makanan yang Anda pesan adalah Bakso");
            break;
          case 2:
            System.out.println("Makanan yang Anda pesan adalah Mie Ayam");
            break;
          case 3:
            System.out.println("Makanan yang Anda pesan adalah Bihun");
            break;
          default:
            System.out.println("Makanan yang Anda pesan diluar menu kami.");
            break;
         }

        switch(minuman){
          case 1:
            System.out.println("Minuman yang Anda pesan adalah Es Campur");
            break;
          case 2:
            System.out.println("Minuman yang Anda pesan adalah Soda");
            break;
          case 3:
            System.out.println("Minuman yang Anda pesan adalah Air Putih");
            break;
          default:
            System.out.println("Minuman yang Anda pesan diluar menu kami.");
            break;
         }

    }
  }
```

P2.4 Daftar Pustaka

Naughton, Patrick, *Java Handbook: Konsep Dasar Pemrograman Java*, Andi Yogyakarta, 1996.

Gary Cornell dan Cay S.Horstmann, *Core Java edisi Indonesia*, Andi, Yogyakarta, 1997.

ANuff, *Penuntun Pemrograman Java*, Andi Yogyakarta, 1997.

Abdul Kadir, *Dasar Pemrograman Java 2*, Andi Yogyakarta, 2008.